Problem C: Playlist Order

Mr. Tortoise's living room. Achilles and the Tortoise are enjoying a cup of tea. A computer nearby is playing beautiful music from a YouTube playlist.

Achilles: I've got to say, Mr. T, this has been a delightful evening. And you have such a refined taste in music!

Tortoise: Thank you, Achilles. This is one of my favourite music playlists. How nice to have YouTube to play music like this.

Achilles: Yes, and don't forget videos of cats. Although, I feel something's a bit funny. I have listened to this playlist many times before, but, did you shuffle the order of the songs?

Tortoise: Good observation. It just so happens that, from time to time, I like to play my lists in different orders. For instance, this particular playlist we're listening to has 5 songs in it, and I call this arrangement 3-2-1-1-0.

Achilles: I see. But what is the meaning of that sequence of numbers? I don't get it.

Tortoise: Well, as you know, Achilles, I am very fond of discrete mathematics. I like identifying a particular playlist order by using an *inversion table*, which is a mechanism to encode a particular permutation of a sequence of numbers.

Achilles: That sounds interesting, and I don't find it surprising coming from you, Mr. T. But would you please elaborate; how does an inversion table work?

Tortoise: Of course. Imagine a sequence of the first N positive integers: $1, 2, \ldots, N$. Such a sequence can be used to describe the original order in a playlist with N songs. Now, imagine a sequence of numbers $s_1, s_2, \ldots, s_N$. Every number s_i in this sequence (with $1 \le i \le N$) simply represents the number of integers greater than i that are found *to the left* of i in a specific permutation of the original sequence.

Achilles: Give me a minute… Aha! I think I understand. So the arrangement we're listening to, 3-2-1-1-0, means that there are exactly 3 songs that came originally *after* song #1, but in this permutation now come *before* it. Song #2 is preceded by 2 songs that were originally *after* it, and so on.

Tortoise: That's exactly right. So, tell me, Achilles. Do you think you can determine the specific permutation we're listening to if its inversion table is 3-2-1-1-0?

Achilles: I know I can! Let's see…

Help Achilles determine the permutation that corresponds to the given inversion table for the first N positive integers.

Input

Input starts with a positive integer **T**, that denotes the number of test cases.

Each test case is described in two lines. The first line contains a single integer N. The second line contains a list of N integers, describing the inversion table $s_1, s_2, \ldots, s_N$.

$T \le 1000$; $1 \le N \le 100$; $0 \le s_i < N$

Output

For each test case, print the case number, followed by the permutation of numbers. If the inversion table is invalid —it doesn't describe a correct permutation—, then print **impossible**.

Sample Input	Output for Sample Input

3
5
3 2 1 1 0
3
1 2 0
3
0 0 0
	Case 1: 5 3 2 1 4
	Case 2: impossible
	Case 3: 1 2 3